

# Animation de plis de tissus pour le rendu non-photoréaliste

Master Recherche Image Vision Robotique  
Promotion 2005/2006, Mars-Juin 2006

**Thierry STEIN**

Encadrants : Joëlle THOLLOT et François FAURE

ARTIS - EVASION/ GRAVIR, INRIA Rhône-Alpes - ZIRST

655 avenue de l'Europe  
38334 Saint Ismier Cedex



CENTRE NATIONAL  
DE LA RECHERCHE  
SCIENTIFIQUE



INRIA  
RHÔNE-ALPES



UNIVERSITÉ  
JOSEPH FOURIER  
SCIENCES TECHNOLOGIE MÉDECINE



# Remerciements

Je tiens d'abord à remercier chaleureusement mes deux encadrants François Faure et Joëlle Thollot. Pendant ces quelques trois mois, j'ai pu bénéficier de leur gentillesse, leur disponibilité, leurs conseils et leur expérience. Ils resteront pour moi un peu plus que de simples encadrants.

Je remercie l'ensemble des équipes ARTIS et EVASION de m'avoir accueilli parmi eux, et notamment leurs directeurs François Sillion et Marie-Paule Cani. Je remercie aussi l'ensemble des permanents, avec une pensée particulière pour Franck (à quand la belle ?).

Je pense aussi à tous ceux qui ne m'ont pas fait regretter d'avoir effectué cette année d'études : l'ensemble des membres du DEA, et plus particulièrement Grégoire, Adrien, Grégory, Nicolas, Thibault, Rishi (on en aura gagné un :-).

Enfin, un immense merci à Evelyne, sans qui rien ne serait arrivé.



# Table des matières

<b>1</b>	<b>Présentation</b>	<b>7</b>
1.1	Vêtements, plis et dessins . . . . .	7
1.2	Vêtements et synthèse d'images . . . . .	8
1.3	Le problème du rendu non-photoréaliste . . . . .	9
1.4	Une approche "A l'envers" . . . . .	9
<b>2</b>	<b>État de l'art</b>	<b>11</b>
2.1	Qu'entend-on par "Rendu non-photoréaliste" ? . . . . .	11
2.2	Les lignes caractéristiques dans le dessin au trait . . . . .	12
2.2.1	Silhouettes . . . . .	12
2.2.2	Contours suggestifs . . . . .	13
2.2.3	Crêtes et vallées . . . . .	13
2.3	L'animation de vêtements . . . . .	14
2.3.1	Modèles physiques et calculs numériques . . . . .	14
2.3.2	Représentation du phénomène émergeant . . . . .	15
2.4	Intérêt d'une méthode directe . . . . .	17
<b>3</b>	<b>Travail réalisé</b>	<b>19</b>
3.1	Principe général . . . . .	19
3.2	Approche par triangles . . . . .	19
3.2.1	Calcul des déformations . . . . .	19
3.2.2	Dessin des lignes sur le maillage . . . . .	23
3.2.3	Bilan . . . . .	25
3.3	Approche par sommets . . . . .	26
3.3.1	Calcul des déformations . . . . .	26
3.3.2	Dessin des lignes . . . . .	30
3.3.3	Comparaison . . . . .	32
3.4	Animation des plis et cohérence temporelle . . . . .	33
3.5	Rendu . . . . .	34
3.5.1	Rendu calligraphique simple . . . . .	34
3.5.2	Carte de déplacement . . . . .	36
<b>4</b>	<b>Bilan</b>	<b>37</b>
4.1	Résultats . . . . .	37
4.2	Bilan . . . . .	41
4.3	Perspectives . . . . .	41



# Chapitre 1

## Présentation

### 1.1 Vêtements, plis et dessins

Au même titre que le nu, le dessin de drapé est un exercice de style classique pour tout apprenti-dessinateur. Le terme "drapé" englobe tout un ensemble de plis, que ce soit dans des tentures, des rideaux ou des tissus divers. En ce sens, la draperie est une composante essentielle de notre environnement visuel.



FIG. 1.1 – *Quatre études de draperies* Anonyme, vers 1730-1735

Les plis véhiculent généralement une information de mouvement. Ainsi, dans l'exemple de la figure 1.2, le drapé de la robe de Falordelle nous aide à imaginer la danse qu'elle effectue autour du feu. Au même titre que le feu, la chevelure ou encore les notes de musique, cet indice

visuel nous communique l'illusion d'une case vivante et dynamique alors que nous sommes en face d'une image fixe.



FIG. 1.2 – Falordelle dansant autour du feu, extrait de [AT99]

## 1.2 Vêtements et synthèse d'images

La simulation de tissus est un domaine très actif de la recherche en synthèse d'image depuis maintenant une vingtaine d'années. Depuis Weill en 1986, le problème a été abordé sous toutes les coutures et fournit aujourd'hui, dans le cadre d'animation off-line, des résultats d'une qualité remarquable (figure 1.3). Par contre, dans le domaine des applications interactives, ce problème reste encore largement ouvert. Les rares algorithmes prévus pour fonctionner en temps réel peinent encore à simuler de manière crédible le phénomène de formation des plis, un détail pourtant crucial pour le réalisme visuel.



FIG. 1.3 – Geri, image extraite de [DKT98]



### 1.3 Le problème du rendu non-photoréaliste

Comme nous le verrons dans l'état de l'art, le rendu non-photoréaliste (NPR<sup>1</sup>) ajoute une dimension supplémentaire à notre problème, à savoir la nécessité de calculer les lignes importantes au niveau visuel. Par lignes importantes, on entend celles qui sont communément représentées par un dessinateur, à l'exclusion des ombrages tels que les hachures. Dans le cas des plis de tissus, ces lignes sont généralement les crêtes et les vallées qui se forment sur la surface du maillage. Ce calcul est encore aujourd'hui brité du fait de la difficulté de calculer une courbure, notion par essence continue, sur une surface discrète. De plus, pour qu'un tel calcul soit pertinent, on a besoin de mailler très finement le tissu, ce qui alourdit naturellement le temps de simulation pour un pas d'animation.



FIG. 1.4 – Un exemple de plis "non-photoréalistes". Extrait du site [www.packrabbitpress.com](http://www.packrabbitpress.com)

### 1.4 Une approche "A l'envers"

Le processus classique pour simuler des plis en NPR se déroule en deux étapes. Dans un premier temps, on doit déformer la géométrie du tissu, pour venir dans un deuxième temps extraire les lignes caractéristiques correspondant aux plis apparus.

Il existe aujourd'hui des techniques pour réaliser la première étape en temps interactif, mais la deuxième reste encore un problème. Ceci vient du fait que si chacune de ces questions a été largement étudiée de manière isolée, il n'existe pour l'instant à notre connaissance aucune méthode s'attaquant directement au cas des plis pour le NPR.

Il nous semble qu'une approche globale est pourtant possible et réalisable en temps interactif si on décide de prendre le problème "à l'envers". En clair, nous pensons qu'une bonne manière de procéder est de commencer par déterminer les lignes de plis caractéristiques qui se forment au cours de l'animation, puis d'utiliser ces lignes comme support pour venir déformer la géométrie du tissu. Une telle approche a plusieurs avantages : elle assure que les crêtes et les vallées sont correctes sans avoir à calculer de courbure tout en permettant d'obtenir des temps de calcul de l'ordre de plusieurs images par seconde.

---

<sup>1</sup> pour Non Photorealistic Rendering

La suite de ce document s'organise de manière classique. L'état de l'art (chapitre 2) revient sur les notions de lignes caractéristiques et les techniques d'animation de tissus. Le chapitre 3 expose notre algorithme, les résultats qu'il fournit étant présentés et discutés au chapitre 4.

## Chapitre 2

# État de l'art

Comme il n'existe pas à notre connaissance de méthode dédiée spécifiquement à notre problème, ce chapitre présente de manière distincte les résultats actuels en animation de vêtements et rendu non-photoréaliste. On s'intéressera tout particulièrement aux méthodes interactives de simulation de plis, en essayant de voir pourquoi elles échouent dans le cadre du NPR.

### 2.1 Qu'entend-on par "Rendu non-photoréaliste" ?

Éthymologiquement parlant, le rendu-non photoréaliste se définit par une négation. Il s'oppose au rendu photoréaliste d'une image, qui vise à reproduire par ordinateur la réalité de la manière la plus juste possible. Le NPR est apparu depuis quelques années pour répondre à une demande différente. Formellement, il consiste à mettre en évidence certains détails d'une scène 3D, par l'intermédiaire d'une étape de simplification, de manière à communiquer l'essentiel de l'information rapidement et efficacement. Ceci trouve des applications dans de nombreux domaines : dessin technique ou CAO, message visuel, industrie médicale, ou même plus simplement dans un intérêt esthétique. Les exemples de la figure 2.1 montrent la multi-

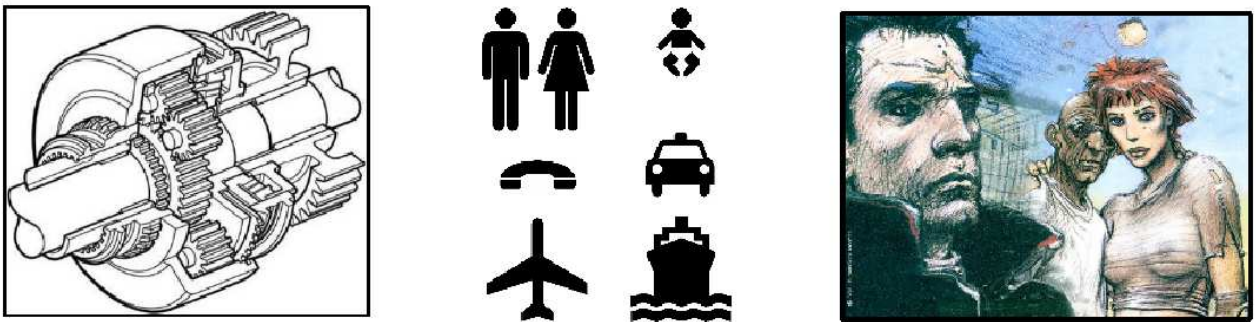


FIG. 2.1 – Exemples d'images non-photoréalistes

plicité de formes que peut prendre le rendu non-photoréaliste. On trouvera une présentation

complète des différentes techniques de NPR dans les livres de Bruce & Amy Gooch [GG01], et de Thomas Stothotte et Stefan Schlechtweb [SS02].

Le travail présenté dans ce document s'intéresse plus précisément au rendu non-photoréaliste de dessin au trait à partir d'une scène 3D, branche qui consiste à extraire, à partir d'un modèle 3D, un ensemble de lignes caractéristiques définissant la forme et le contenu de la scène.

## 2.2 Les lignes caractéristiques dans le dessin au trait

Cette section présente les différentes techniques existantes permettant d'extraire des lignes caractéristiques à partir d'un modèle 3D. Ces lignes n'ont pas d'existence physique réelle, elles correspondent aux limites que nous percevons dans les formes par le biais de variations de ton et de texture. De nombreux chercheurs se sont penchés sur l'identification mathématique de ces lignes. Ceci reste bien évidemment subjectif, dans la mesure où chaque artiste choisit les traits qu'il dessine en fonction de sa perception personnelle de la scène qu'il cherche à représenter. Cependant, un consensus se dégage sur trois types de lignes qui sont les plus couramment utilisées en NPR : les silhouettes, les contours suggestifs, et les crêtes et vallées.

### 2.2.1 Silhouettes

Ce sont les lignes les plus communément représentées dans les dessins. La silhouette d'une surface correspond aux endroits de cette surface où celle-ci passe d'une position où elle est orientée face au point de vue à une position où elle est orientée dans l'autre sens.

Mathématiquement, prenons une surface  $S$  vue depuis un point de vue  $v$ . Un point  $p$  de  $S$  appartient à la silhouette si et seulement si :

$$(p - v) \cdot \vec{n} = 0 \quad (2.1)$$

$\vec{n}$  étant la normale à  $S$  au point  $p$ . L'ensemble des points  $p$  tels que  $(p - v) \cdot \vec{n} > 0$  correspond aux points où  $S$  est orientée face au point de vue tandis que l'ensemble des points  $p$  tels que  $(p - v) \cdot \vec{n} < 0$  correspond aux points où  $S$  est orientée dans l'autre sens. La silhouette est donc dépendante du point de vue.

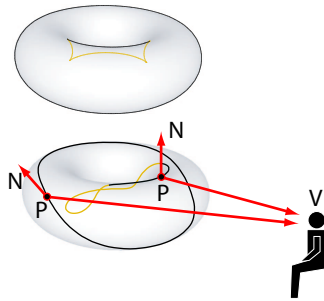


FIG. 2.2 – Définition de la silhouette

Parmi les travaux sur les silhouettes, citons celui de Kalnins et al [KDMF03] qui permet d'animer en temps réel une silhouette stylisée.

### 2.2.2 Contours suggestifs

Les contours suggestifs ont été introduits en 2003 par DeCarlo [DFRS03, DFR04]. Ils correspondent intuitivement aux points qui sont des “presque silhouettes”, c’est-à-dire des points qui appartiendront à la silhouette sous des points de vue voisins. Les contours suggestifs peuvent être combinés aux contours de silhouette dans la mesure où ils les étendent ou les complètent (voir figure 2.3). Ils sont définis comme l’ensemble des points de la surface pour lesquels la courbure radiale change de signe et augmente en se rapprochant du point de vue. Formellement, les contours suggestifs sont définis par

$$P = \left\{ x \in S \mid K_r(x) = 0; \frac{\partial K_r(x)}{\partial w(x)} > 0 \right\}$$

Où  $K_r$  est la courbure radiale (c’est-à-dire la courbure dans la direction de vue),  $w$  la projection du vecteur de vue sur le plan tangent en  $x$ , et  $S$  la surface de l’objet représenté.

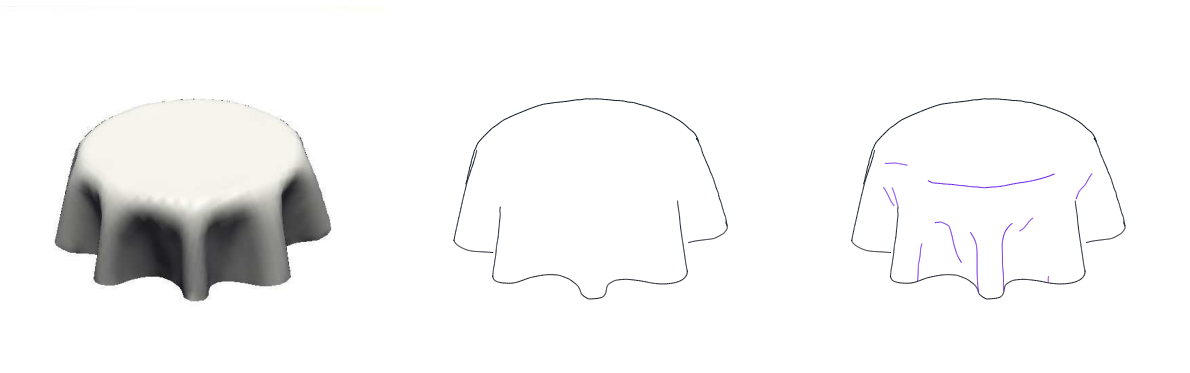


FIG. 2.3 – A gauche le modèle, au milieu les contours et à droite les contours suggestifs en violet.

### 2.2.3 Crêtes et vallées

Les crêtes et vallées [YBS05, OBS04, CP05] sont des lignes saillantes sur un objet. Elles font ressortir les zones de changement dans la forme du modèle.

Les crêtes (respectivement vallées) correspondent au lieu des maxima locaux de courbure dans la direction de courbure principale maximale (respectivement minimale). Les crêtes se trouvent donc sur les zéros de la dérivée de la courbure dans la direction principale maximale. Formellement, le lieu des crêtes vérifie :

$$e_1 = 0, \quad \frac{\partial e_1}{\partial t_1} < 0, \quad K_1 > |K_2|$$

et pour les vallées :

$$e_2 = 0, \quad \frac{\partial e_2}{\partial t_2} > 0, \quad K_2 < -|K_1|$$

Avec  $K_1, K_2$  les courbures principales maximale et minimale,  $t_1, t_2$  les directions de courbures principales maximale et minimale,  $e_1 = \frac{\partial K_1}{\partial t_1}$ , et  $e_2 = \frac{\partial K_2}{\partial t_2}$ .

Les lignes extraites sont indépendantes du point de vue et uniquement basées sur la géométrie du modèle. Comme ces lignes se basent sur la courbure, il faut faire attention à l'estimation de la courbure sur le maillage, et souvent on obtient un certain nombre de lignes causées par des erreurs d'approximation. Il est donc nécessaire de trouver un bon seuil pour ne garder que les crêtes et vallées porteuses d'information (voir figure 2.4).

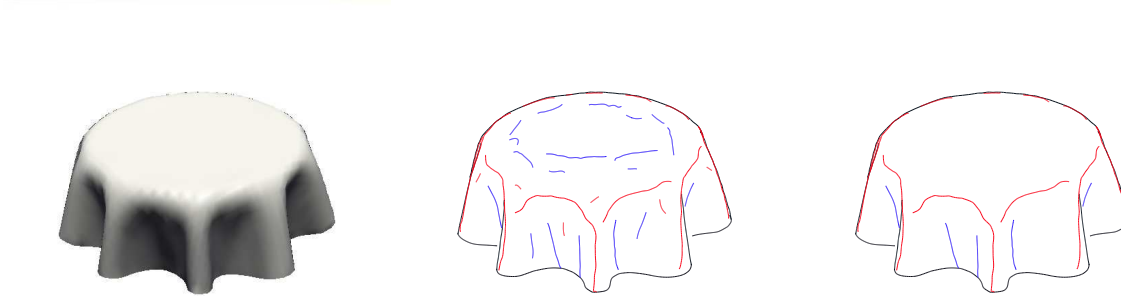


FIG. 2.4 – Crêtes en rouge, vallées en bleu. A droite : seuil de 0, à gauche seuil de 2.

Les plis de vêtements, et plus généralement de tissus, correspondent à ce type de ligne, comme on peut le constater sur la nappe (figure 2.4). Leur calcul nécessite donc que le maillage de notre modèle soit déformé au cours de l'animation. Ceci induit certains problèmes, comme expliqué dans la section suivante.

## 2.3 L'animation de vêtements

On peut catégoriser en deux groupes les recherches menées depuis un quart de siècle en animation de tissus : celles qui cherchent à reproduire le comportement du tissu en résolvant des systèmes numériques issus de la physique ; et celles cherchant à modéliser le phénomène émergent et à le reproduire à partir de quelques règles issues de l'observation. Cette deuxième approche sera largement mise en avant dans la suite de ce chapitre car elle nous semble la plus prometteuse pour réaliser une simulation de qualité en un temps interactif.

### 2.3.1 Modèles physiques et calculs numériques

L'animation de vêtements utilisant des modèles physiques est maintenant relativement standardisée et maîtrisée. Concrètement, les algorithmes actuels partent de l'équation fondamentale de la dynamique :

$$\ddot{x} = M^{-1}(-\frac{\partial E}{\partial x} + F) \quad (2.2)$$

avec

- $x$  le vecteur définissant l'état géométrique du système.
- $M$  une matrice diagonale décrivant la distribution de masse du vêtement.
- $E$  un scalaire fonction de  $x$  représentant l'énergie interne du tissu.
- $F$  fonction de  $x$  et  $\dot{x}$ , les autres forces agissant sur le vêtement (la gravité par exemple).

Cette équation différentielle est ensuite résolue à l'aide de méthodes d'intégration numérique, explicites ou implicites. On doit enfin détecter et traiter les collisions et auto-collisions qui peuvent se produire au cours du pas de temps.

Parmi les résultats importants de ces dernières années, citons les travaux de Baraff et Witkin ([BW98],[BWK03]), Bridson ([BFA02], [BMF03]), et Choi et Ko ([CK02]). Si les résultats sont aujourd'hui d'une bonne qualité visuelle (par exemple Figure 2.5), ces méthodes ne peuvent pas encore s'appliquer dans le cas d'une simulation interactive. Elles nécessitent de plus, dans le cas du NPR, un post-traitement de détection de crêtes et vallées, étape que nous souhaitons éviter.

On trouvera de nombreux détails sur ce pan de la recherche dans le livre de Donald H. House et David E. Breen [HB00], et dans celui de Pascal Volino et Nadia Magnenat-Thalmann [VMT01].



FIG. 2.5 – Exemple de plis obtenus avec la méthode de Choi et Ko. [CK02]

### 2.3.2 Représentation du phénomène émergent

Les méthodes cherchant à simuler la formation des plis sans passer par un modèle physique sont variées, la rigueur scientifique étant remplacée par l'observation et l'imagination. De nombreuses approches ont été proposées pour des résultats divers. Notre intention n'est pas de toutes les mentionner mais de ne retenir que celles dont l'idée se rapproche de notre algorithme.

Volino et al. [VMT99] proposent une technique rapide qui utilise la variation de longueur des arêtes du maillage pour moduler l'amplitude d'un schéma de rides donné sur chaque triangle. Le maillage est raffiné automatiquement si nécessaire. Le rendu utilise soit du *bump mapping* (simulation de relief par perturbation des normales [Bli78]), soit du *displacement mapping* (perturbation de la surface à partir d'une texture dite de déplacement [Coo84]). Ce travail a ensuite été étendu par Hadap et al. [HBVMT99], en se basant non plus sur les déformations des arêtes mais sur celles des triangles. Le résultat est de très bonne qualité au niveau visuel (figure 2.6). Le problème se situe plutôt au niveau de l'utilisateur qui doit dessiner un motif de rides complet et régler des paramètres peu intuitifs.

Dans le même esprit que Volino et Hadap, Masaki Oshita et al. [OM01] proposent d'utiliser un petit nombre de particules pour simuler la dynamique du vêtement. Ils lissent ensuite la surface au moyen de carreaux de Bézier. Lors de cette étape, ils utilisent la compression des arêtes pour faire apparaître des plis sur le vêtement. Malheureusement, ces derniers semblent



FIG. 2.6 – Exemple de plis obtenus avec la méthode de Hadap et al. [HBVMT99]

se former d'une manière un peu aléatoire. De plus, ils n'illustrent leur méthode que sur l'exemple d'une jupe, en ne donnant aucune garantie pour d'autres types de vêtements tel qu'un pull ou un pantalon. La figure 2.7 montre un exemple de leurs résultats.

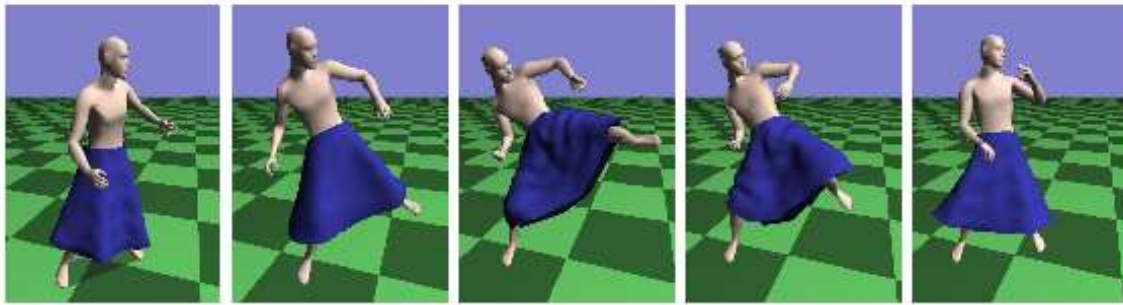


FIG. 2.7 – Exemple de plis obtenus avec la méthode de Oshita et al. [OM01]

On retrouve cependant l'idée de décomposer l'animation du vêtement en deux phases : une simulation dynamique sur un faible nombre de particules, suivie d'un ajout de détails lors du raffinement du maillage pour le rendu. Frédéric Cordier, dans [CMT04], applique cette méthodologie pour créer un système à base de capture vidéo. L'inconvénient de cette approche est l'absence de contrôle sur le résultat final. De plus, lorsque le vêtement se retrouve dans un état trop éloigné des positions de la vidéo, on manque de garantie sur le résultat.

Dans [DJW<sup>+</sup>06], Philippe Decaudin propose une méthode relativement originale qui s'appuie sur les propriétés de flambage des tissus. Constatant que, sur les vêtements approchables par un cylindre, des plis en diamant se forment sous l'effet de la compression, il propose de représenter la surface par un maillage en losanges. Cette approche est rapide et les plis obtenus très réalistes (Figure 2.8). Elle est cependant limitée à certains types de vêtements, ne



s'appliquant pas dans le cas d'une cape par exemple. De plus, la variété des motifs observables reste pour l'instant limitée.

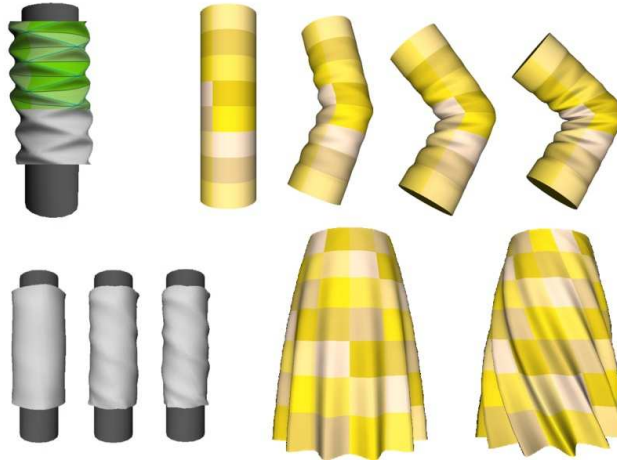


FIG. 2.8 – Exemple de plis obtenus avec la méthode de Decaudin et al. [DJW<sup>+</sup>06]

## 2.4 Intérêt d'une méthode directe

Quels enseignements retirer de ce rapide tour d'horizon ?

Il vient un premier constat évident : pour espérer une simulation interactive, il est nécessaire de décomposer l'animation de tissus en deux étapes. Une première phase, physique, anime une version grossière du vêtement ; puis une deuxième phase, plus géométrique, vient ajouter du détail localement. C'est cette seconde étape qui est l'objet de nos recherches.

La question que nous nous posons est donc la suivante : comment venir dessiner de manière rapide des plis sur un maillage grossier, en gardant une certaine cohérence physique ? Pour effectuer cela, nous avons besoin de certaines informations : où se forment les plis, et dans quelles directions les lignes partent-elles. Nous devons ensuite mettre au point un algorithme d'animation de ces plis, c'est-à-dire définir leur comportement au cours du temps. Finalement, nous devons effectuer un rendu rapide et efficace pour la qualité visuelle.



# Chapitre 3

## Travail réalisé

### 3.1 Principe général

Commençons par préciser les trois objectifs que nous nous fixons pour ce stage :

- Utiliser un maillage relativement grossier, c'est-à-dire que le vêtement est un système masse-ressort contenant un petit nombre de particules.
- Se limiter à une information cinématique simple.
- Dessiner directement les lignes de plis sans calcul de courbure.

Pour cela, nous considérons disposer en entrée d'une position au repos de notre tissu. Au cours de l'animation, nous estimons la déformation subie par le maillage par comparaison entre l'état courant et l'état au repos. Ceci nous fournit une matrice de déformation de laquelle nous extrayons les valeurs et les vecteurs propres. Nous venons ensuite dessiner nos lignes dans les zones de fortes déformations, c'est-à-dire où les valeurs propres sont élevées, dans la direction des vecteurs propres correspondants. Nous obtenons des suites de segments qui sont ensuite lissées avant d'être affichées.

Nous présentons deux méthodes pour calculer la déformation. La première estime la déformation d'un triangle à partir des déplacements de ses sommets ; la seconde estime la déformation d'un sommet en fonction des déplacements de ses voisins. La méthode choisie pour estimer la déformation modifie légèrement la manière de dessiner nos lignes de plis, nous discuterons donc des avantages et des inconvénients de ces deux approches.

Nous présentons ensuite l'algorithme mis en place pour animer les lignes de plis. Ce dernier assure une cohérence temporelle dans l'évolution des lignes, ce qui ne serait pas le cas avec une approche naïve. Nous terminerons par quelques remarques sur le rendu de ces lignes.

### 3.2 Approche par triangles

#### 3.2.1 Calcul des déformations

Pour calculer les déformations d'un maillage, nous utilisons ici la méthode développée par Matthieu Nesme et al [NPF05]. On part d'une position au repos de notre tissu. Pour chaque sommet du maillage, on dispose d'une position initiale définie par son vecteur position  $X_0$ . Au cours du mouvement, chacun de ses points se retrouve à une position différente  $X$ . Le déplacement  $u$  correspond au vecteur de la différence entre l'état déformé et l'état au repos

d'une particule  $x$  :

$$u(x) = X - X_0 = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}$$

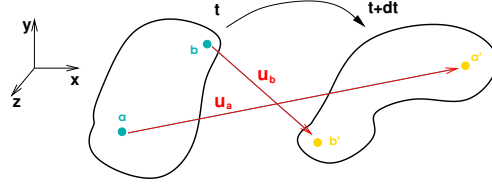


FIG. 3.1 – Déplacements  $u_a$  et  $u_b$  des particules  $a$  et  $b$

Les déformations s'obtiennent ensuite à partir des déplacements grâce à un tenseur de déformations qui fait intervenir la géométrie du milieu. Les déformations représentent les dilatations (élongation, compression) et les cisaillements que subit un objet (figure 3.2).

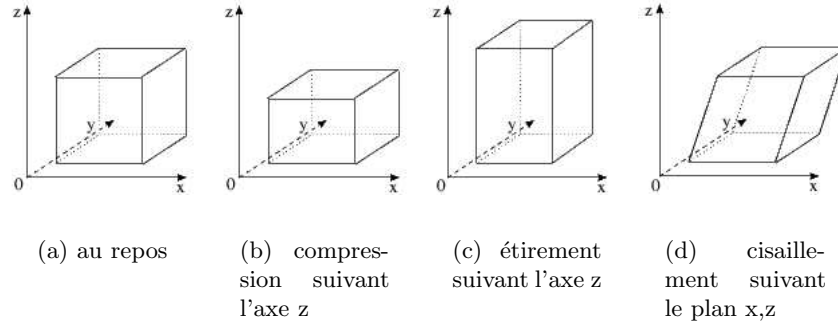


FIG. 3.2 – Les différentes déformations d'un hexaèdre

Pour chaque élément, la formulation de la déformation est linéaire. Elle s'écrit sous la forme :

$$d = Bu$$

- $u$  est le vecteur déplacement défini plus haut.
- $B$  est la matrice de rigidité de l'élément. Elle s'écrit :

$$B = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & 0 & \dots & \frac{\partial N_n}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \dots & 0 & \frac{\partial N_n}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial z} & \dots & 0 & 0 & \frac{\partial N_n}{\partial z} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & 0 & \dots & \frac{\partial N_n}{\partial y} & \frac{\partial N_n}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial z} & \frac{\partial N_1}{\partial y} & \dots & 0 & \frac{\partial N_n}{\partial z} & \frac{\partial N_n}{\partial y} \\ \frac{\partial N_1}{\partial z} & 0 & \frac{\partial N_1}{\partial x} & \dots & \frac{\partial N_n}{\partial z} & 0 & \frac{\partial N_n}{\partial x} \end{bmatrix}$$

où  $N_i$  est la fonction d'interpolation du  $i^{eme}$  noeud et  $n$  le nombre de noeuds de l'élément. Les fonctions d'interpolation dépendent du type de l'élément. Dans le cas présent, on se réduit à des triangles.

En pratique, on effectue le calcul des déformations en le décomposant en une déformation pure et une rotation. Pour cela, on passe par un élément unitaire, comme montré dans la Figure 3.3.

On obtient ainsi, pour chaque triangle, une matrice de déformation de la forme :

$$\varepsilon = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} \\ \varepsilon_{xy} & \varepsilon_{yy} \end{bmatrix}$$

$\text{varepsilon}_{xx}$  correspond à la déformation selon l'axe  $x$ ,  $\text{varepsilon}_{yy}$  correspond à la déformation selon l'axe  $y$ , et  $\text{varepsilon}_{xy}$  correspond au cisaillement. Cette matrice est réelle et symétrique, elle est donc diagonalisable. On peut alors calculer ses valeurs et ses vecteurs propres, ce qui donne pour chaque triangle deux valeurs de déformation avec leurs directions associées (voir figures 3.4 et 3.5).

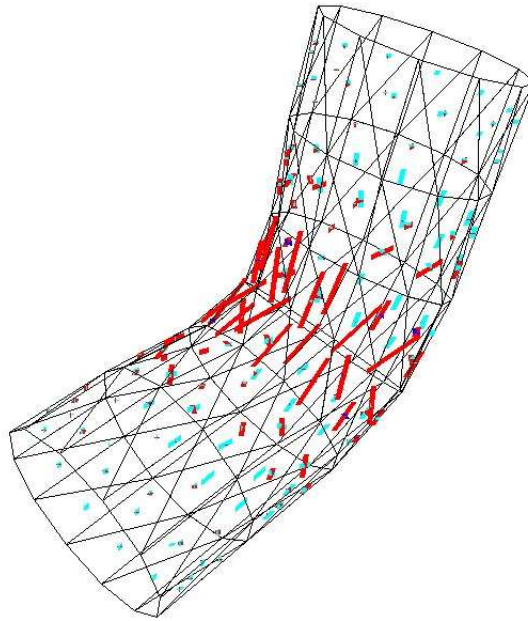
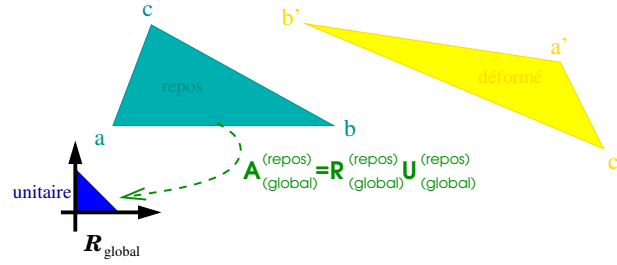
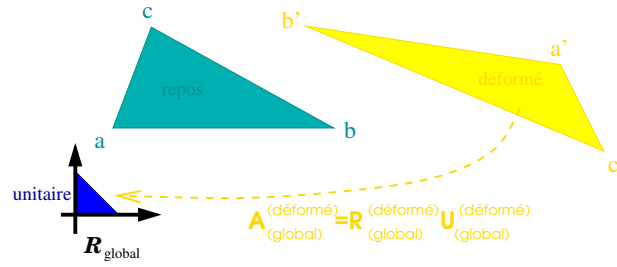


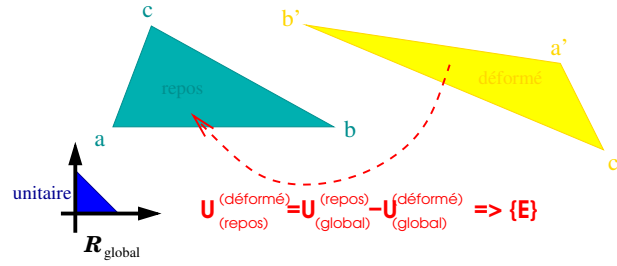
FIG. 3.4 – Exemple de vecteurs propres pour un cylindre qui se plie, la longueur des vecteurs étant fonction de la valeur propre.



(a) étape 1 : on précalcule la déformation de l'élément au repos vers l'élément unitaire



(b) étape 2 : on calcule la rotation et la déformation de l'élément déformé vers l'élément unitaire



(c) étape 3 : on obtient la déformation entre l'élément déformé et l'élément au repos en passant par l'élément unitaire

FIG. 3.3 – Calcul des déplacements

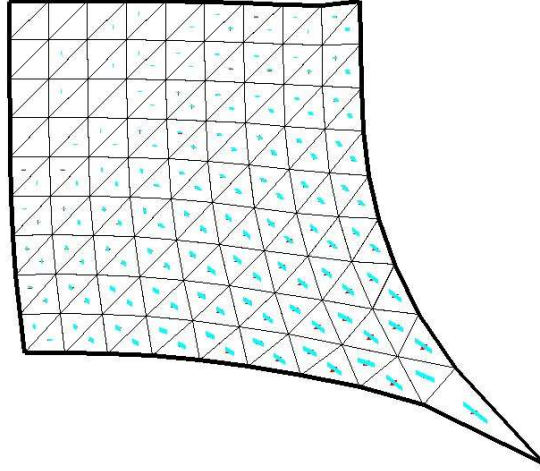


FIG. 3.5 – Exemple de valeurs et vecteurs propres pour un drap tiré vers le bas.

### 3.2.2 Dessin des lignes sur le maillage

Les valeurs et vecteurs propres calculés dans la section 3.2.1 nous donnent des informations sur la localisation et l'orientation des lignes de plis que nous souhaitons tracer. Nous utilisons ces informations pour dessiner nos lignes de la manière suivante : nous choisissons comme point de départ le barycentre du triangle qui est le plus déformé (Figure 3.6). Sur ce triangle, nous traçons une ligne (Figure 3.7) passant par le barycentre et dans la direction :

- du vecteur propre pour les extensions (déformation positive)
- orthogonale au vecteur propre pour les compressions (déformation négative)

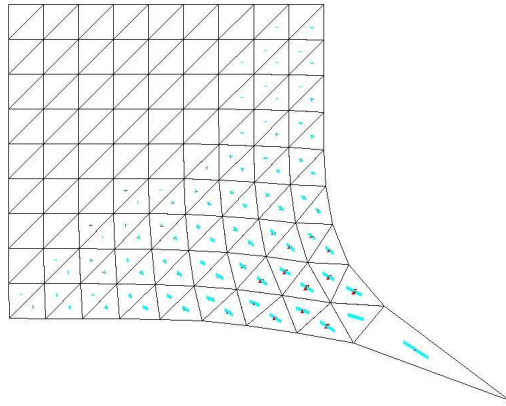


FIG. 3.6 – étape 1 : on choisit un point de départ.

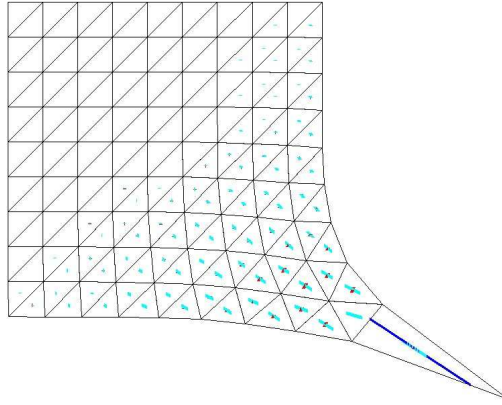


FIG. 3.7 – étape 2 : on dessine notre trait en fonction du vecteur propre.

Dans les cas non dégénérés, ce segment intersecte deux arêtes du triangle. On continue donc le processus en passant aux triangles voisins (Figure 3.8), en modifiant la direction de tracé en fonction des vecteurs propres. Ceci se fait en deux passes, une pour chaque intersection entre notre ligne de plis et le triangle de départ. On itère ce processus jusqu'à l'une des conditions d'arrêt suivantes (Figure 3.9) :

- La ligne forme une boucle, c'est-à-dire qu'elle revient sur un triangle que l'on a déjà traversé ;
- La déformation du triangle est trop faible (*ie* inférieure à un seuil numérique paramétrable) ;
- Il n'y a pas de triangle voisin (*ie* on est sur un bord du maillage).

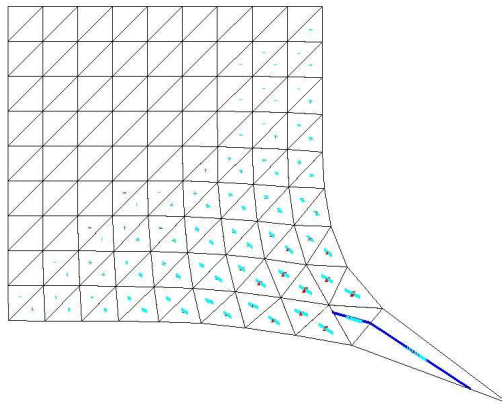


FIG. 3.8 – étape 3 : on passe aux triangles voisins.



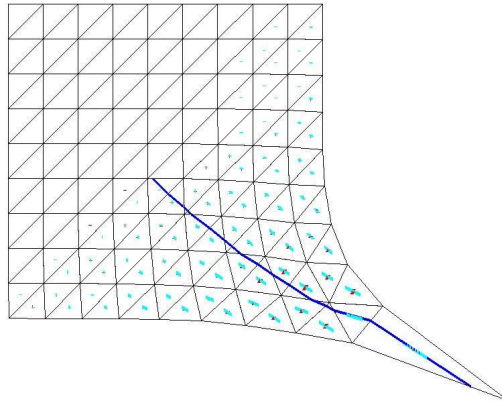


FIG. 3.9 – étape 4 : on trace jusqu'à une des conditions d'arrêt.

On peut ensuite réitérer ce processus pour tracer autant de lignes que l'on souhaite. Pour éviter que toutes les lignes de plis viennent s'agglomérer au même endroit, on définit le voisinage d'un pli. Basiquement, le voisinage est l'ensemble des triangles traversés par la ligne. D'autres définitions sont possibles, mais celle-ci se révèle être la plus simple et la plus souple d'usage. En pratique, tous les triangles ayant déjà été traversés sont désactivés, c'est-à-dire que l'on interdit à une autre ligne de passer dans ce voisinage. Ceci impose que seulement une ligne puisse passer par un triangle mais évite que toutes les lignes viennent se condenser dans un seul endroit. Dans le cas d'un maillage fin, on peut étendre le voisinage aux triangles voisins de ceux appartenant à la ligne que nous venons de tracer, pour la même raison de contrôle de densité. Notons enfin que l'on peut autoriser à faire passer plus de lignes par triangle, l'important étant d'éviter qu'elles se retrouvent toutes au même endroit.

Pour itérer le processus, nous ré-effectuons une recherche de maximum dans les triangles autorisés (i.e ceux hors du voisinage de la ou des lignes précédents), ce qui nous donne un nouveau point de départ. L'itération est effectuée tant que le nombre de lignes est inférieur à un nombre fixé, ou bien que tous les triangles restants ont une valeur de déformation trop faible. La figure 3.10 montre un résultat de ce processus.

### 3.2.3 Bilan

Le principal inconvénient de cette méthode est l'évolution trop brutale de nos lignes. Entre l'instant  $t$  et l'instant  $t + 1$ , si une ligne s'allonge, elle le fera en traversant au moins un triangle supplémentaire. Étant donné que nous utilisons un maillage grossier, on voit au cours de l'animation des petits morceaux apparaître ou disparaître à l'extrémité des lignes.

Pour avoir une progression plus douce, nous proposons de calculer la déformation non plus par triangle mais par sommet. Ceci nous permet d'obtenir un champ continu par interpolation linéaire, assurant une évolution beaucoup plus satisfaisante de nos lignes au cours du temps. Ceci est détaillé dans la section suivante.

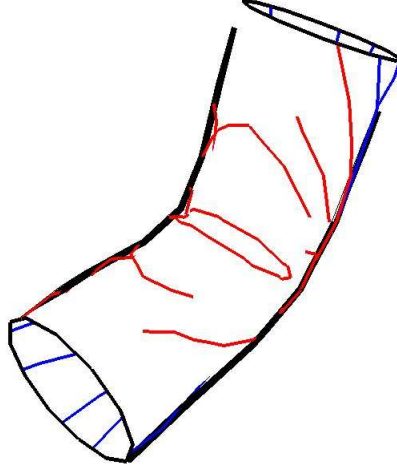


FIG. 3.10 – Plusieurs lignes de plis pour un cylindre plié (silhouette en noir)

### 3.3 Approche par sommets

Dans ce que nous appelons l'approche "par sommets", la principale différence se situe dans le calcul de la déformation. Cependant, les modifications dans le calcul des lignes sont légères

#### 3.3.1 Calcul des déformations

Pour estimer la déformation par sommet du maillage, il faut commencer par définir l'état au repos de ce dernier. Nous représentons l'état au repos de chaque particule par rapport à son voisinage, c'est-à-dire les sommets qui lui sont connectés par une arête. Nous calculons un repère orthonormé local centré sur la particule en définissant ses trois axes de la manière suivante :

- l'axe  $x$  est la projection d'une arête dans le plan tangent à la particule
- l'axe  $z$  est aligné avec la normale à la surface
- l'axe  $y$  se déduit des deux autres par produit vectoriel.

La position au repos d'une particule se définit donc par la position de ses sommets voisins dans son repère local (voir figure 3.11).

A un instant  $t$ , on connaît la position des voisins de la particule dans le repère monde. On calcule alors le nouveau repère local qui minimise la distorsion, c'est-à-dire qui minimise l'écart entre les coordonnées courantes et les coordonnées au repos du voisinage. Soit  $p$  les coordonnées d'une particule de notre maillage. On définit

$$e(0) = \begin{pmatrix} \vec{e}_x(0) \\ \vec{e}_y(0) \\ \vec{e}_z(0) \end{pmatrix}$$

le repère local centré en  $p$  à l'instant initial. Soient  $p_1 \cdots p_n$  les coordonnées dans le repère monde des  $n$  voisins de  $p$  à l'instant initial 0. La matrice

$$J(0) = \begin{pmatrix} p_{1x} - p_x & p_{1y} - p_y & p_{1z} - p_z & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{1x} - p_x & p_{1y} - p_y & p_{1z} - p_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & p_{1x} - p_x & p_{1y} - p_y & p_{1z} - p_z \\ \vdots & & & & & & & & \vdots \\ p_{nx} - p_x & p_{ny} - p_y & p_{nz} - p_z & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{nx} - p_x & p_{ny} - p_y & p_{nz} - p_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & p_{nx} - p_x & p_{ny} - p_y & p_{nz} - p_z \end{pmatrix}$$

représente la position des voisins de  $p$  à l'instant initial. Finalement, le produit  $J(0)e(0)$  représente la position des voisins de  $p$  dans le repère local  $e(0)$ .

On a de même  $e(t)$  le repère local à l'instant  $t$  et  $J(t)$  la position des voisins de  $p$  au même moment. Calculer la déformation du voisinage de  $p$  revient alors à résoudre l'équation matricielle :

$$J(t)e(t) = J(0)e(0) \quad (3.1)$$

On a donc un système de 9 inconnues (le vecteur  $e(t)$ ) à résoudre à chaque pas de temps et pour chaque particule. On peut grandement simplifier ce calcul en faisant les hypothèses suivantes :

- La déformation reste dans le plan tangent, on fixe donc l'axe  $z$  comme étant la normale.
- Les axes  $x$  et  $y$  restent dans le plan tangent.
- L'axe  $x$  reste la projection de la même arête qu'à l'instant initial, et ne peut donc que s'allonger ou se comprimer.

Modulo un déplacement rigide, ceci est équivalent à dire :

$$\begin{aligned} \vec{e}_x(t) &= \alpha \vec{e}_x(0) \\ \vec{e}_y(t) &= \beta \vec{e}_x(0) + \gamma \vec{e}_y(0) \\ \vec{e}_z(t) &= \vec{e}_z(0) \end{aligned}$$

Le système se reformule alors :

$$J(t)(B(t)x + b(t)) = J(0)e(0) \quad (3.2)$$

avec

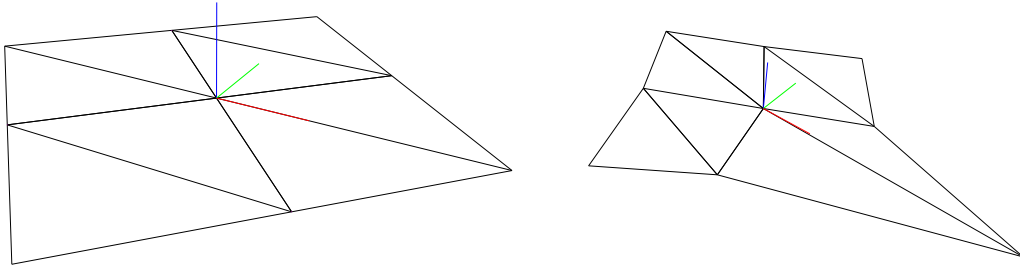
- $x = \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}$  le vecteur des inconnues.
- $b(t) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ e_z(t) \end{pmatrix}$  un vecteur contenant six zéros puis les coordonnées du vecteur normal

à la surface en la particule  $p$  à l'instant  $t$ .

La solution de ce système est classiquement la matrice pseudo-inverse, c'est à dire :

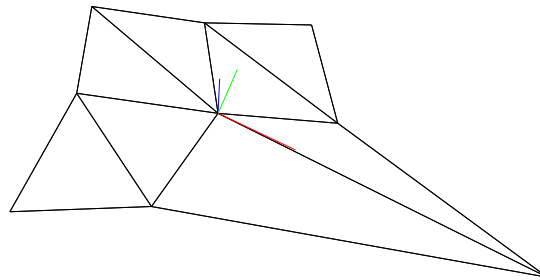
$$x = [(B(t)^T J(t)^T J(t) B(t))]^{-1} B(t)^T J(t)^T (J(0)e(0) - J(t)b(t)) \quad (3.3)$$

A chaque pas de temps, on doit donc inverser la matrice  $[(J(t)B(t))^t J(t)B(t)]$  qui est une matrice  $3 \times 3$ . L'ensemble de ce processus peut se résumer par la figure 3.11.



(a) étape 1 : position au repos du repère local pour la particule central, axe  $e_x$  en rouge, axe  $e_y$  en vert, axe  $e_z$  en bleu.

(b) étape 2 : Le repère local uniquement soumis au déplacement rigide reste ortho-normé.



(c) étape 3 : Les axes  $x$  et  $y$  sont modifiés en fonction des déplacements des particules voisines. Dans cet exemple, l'axe  $x$  est allongé proportionnellement à l'éloignement de la particule en bas à droite.

FIG. 3.11 – Calcul des déformations par sommet

Ce calcul nous donne finalement, pour chaque sommet du maillage, une matrice de déformation de la forme :

$$\varepsilon = \begin{bmatrix} \alpha & \beta/2 \\ \beta/2 & \gamma \end{bmatrix}$$

De la même manière que précédemment, on obtient pour chaque sommet deux valeurs propres de déformation et deux vecteurs propres associés.

L'avantage de cette approche est d'avoir une formulation continue de la déformation en tout point du triangle. En utilisant les coordonnées barycentriques d'un point  $q$  dans un triangle, on peut estimer la déformation en ce point par combinaison linéaire des tenseurs de déformations des trois sommets du triangle auquel il appartient.

Ceci nous donne donc un champ de déformation continu sur l'ensemble de notre maillage, illustré par les figures 3.12 et 3.13.

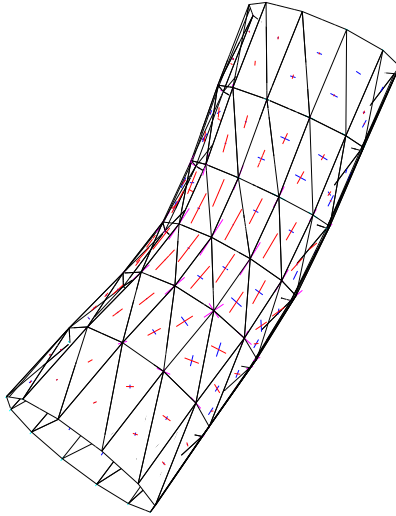


FIG. 3.12 – Exemple de vecteurs propres pour un cylindre qui se plie. En magenta et cyan, les vecteurs aux sommets. En rouge et bleu les vecteurs interpolés aux barycentres des triangles.

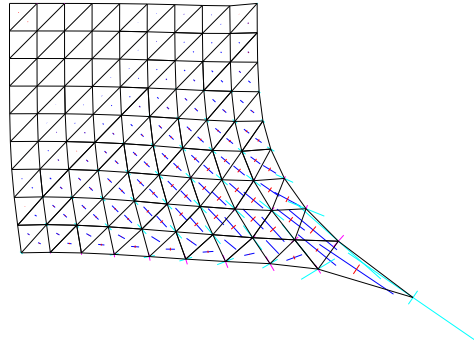


FIG. 3.13 – Exemple de valeurs et vecteurs propres pour un drap tiré vers le bas.

### 3.3.2 Dessin des lignes

Le fait de calculer le champ de déformation par sommet et non plus par triangle modifie notre méthode de dessin de ligne. Afin de prendre en compte la continuité du champ, on fixe un pas d'échantillonnage pour le tracé.

Comme pour la méthode précédente, il faut commencer par choisir un point de départ. Supposons que ce soit le sommet qui subit la plus grande déformation. Nous sommes confrontés à plusieurs inconvénients :

- il est nécessaire de parcourir tous les triangles incidents à ce sommet pour tracer le début de notre ligne.
- on doit ensuite s'assurer que lorsqu'on calcule une autre ligne, le nouveau sommet de départ ne donne pas une ligne trop proche de la précédente.

Nous avons donc par commodité décidé de prendre comme point de départ le barycentre du triangle dont la déformation est la plus grande (figure 3.14). On reste ainsi dans le triangle pour calculer le départ de la ligne tandis que le deuxième inconvénient se résoud de lui même, surtout dans le cadre d'un maillage grossier.

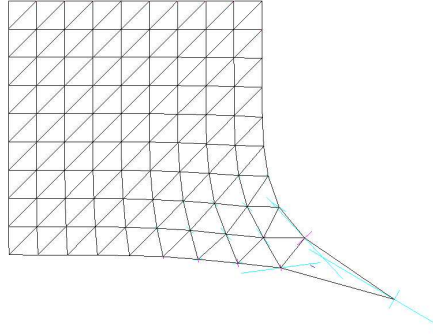


FIG. 3.14 – Choix d'un point de départ.

Une fois ce point de départ choisi, on avance d'une longueur de pas dans la direction de pli. On obtient un nouveau point pour lequel on recalcule la déformation et la direction de pli et ainsi de suite (Figure 3.15). Les conditions d'arrêt sont les mêmes que pour la méthode par triangle : arrivée sur un bord, déformation trop faible, ou formation d'une boucle par notre ligne.

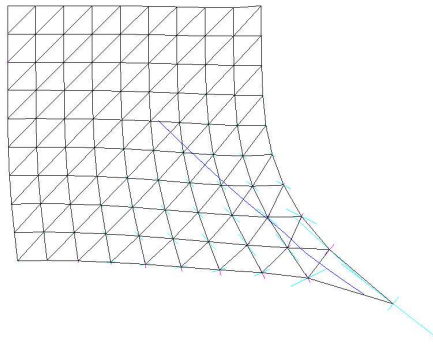


FIG. 3.15 – Itération.

On répète ce processus pour calculer un ensemble de lignes, en autorisant toujours un nombre maximum de lignes à passer dans un triangle. Un exemple de ceci est fourni par la figure 3.16. On remarque que les lignes obtenues sont beaucoup plus lisses que dans la première approche.

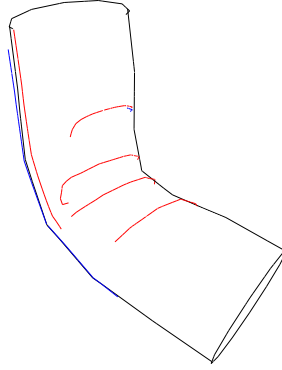


FIG. 3.16 – Exemple d’un ensemble de lignes pour le cylindre.

### 3.3.3 Comparaison

Nous avons présenté deux techniques de calcul du champ de déformation, il est donc intéressant de les comparer. Le calcul de déformation par triangle a l’avantage d’être rapide, comparativement au calcul par sommet. Ceci est cependant pondéré par l’opération de lissage appliquée dans le premier cas, opération nécessaire pour afficher une courbe lisse et non une vilaine suite de segments. De plus, le fait de disposer d’un champ continu donne une souplesse supplémentaire, par exemple pour définir une zone d’influence du pli.

On remarque aussi que les champs obtenus ne sont pas tout à fait similaires, principalement dans les zones de faible déformation. Ceci est probablement dû au fait notre méthode de calcul par sommet est sensible à la structure du maillage. Une solution serait alors de ne pas considérer uniquement les sommets adjacents à une particule mais de prendre également en compte les particules suffisamment proches pour être significatives.

Par exemple, dans le cas trivial de la figure 3.17, on ne considère que les sommets verts comme étant voisins du rouge. Ainsi, la déformation que nous calculons varie en fonction de l’orientation des arêtes. Ceci pourrait être amélioré en prenant en compte les voisins du ”deuxième ordre”, tels que les sommets bleus de l’exemple.

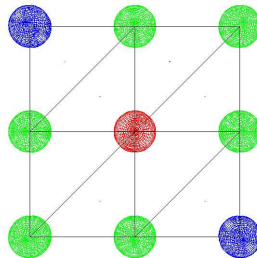


FIG. 3.17 –



### 3.4 Animation des plis et cohérence temporelle

Si nous nous contentons de répéter ces algorithmes à chaque pas de temps, nous distinguons des sauts de ligne qui nuisent à la qualité de l’animation. Cela se traduit par des clignotements, des translations soudaines et de brusques changements de direction. On trouve en figure 3.18 un exemple de ce genre de problème. Une répétition naïve de l’algorithme donne donc ce genre

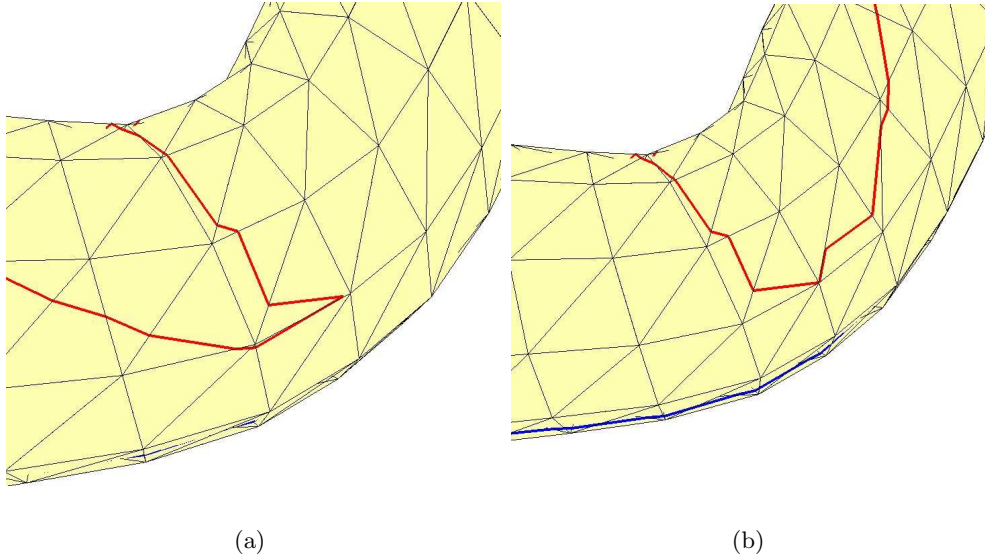


FIG. 3.18 – Changement de direction d’une ligne pour 2 images successives.

de résultat. Une première raison est évidente : le triangle le plus déformé peut changer d’une image à la suivante. On a alors envisagé de mettre en place une stratégie simple : conserver d’une image à l’autre les mêmes points de départ pour nos lignes. Ceci s’est néanmoins révélé insuffisant. En effet, nous n’avons aucune assurance que la ligne va repasser exactement par les mêmes arêtes que précédemment. Il suffit d’un petit changement, comme une légère rotation des directions de déformation, pour entraîner de grandes conséquences au niveau visuel, comme on peut le constater sur la figure 3.18. Ici, en changeant d’arête de sortie, la ligne change complètement d’orientation. Tout ceci est naturellement rédhibitoire dans le cadre d’une animation et nous force à mettre en place une stratégie qui soit robuste à ces artefacts.

L’algorithme que nous avons mis en place se base sur l’idée suivante : au lieu de recalculer nos lignes d’une image à l’autre, nous modifions celles de l’étape précédente en fonction des évolutions du champ de déformation. Celui-ci peut varier de deux manières : en intensité (la compression d’un triangle augmente ou diminue) et en orientation (les vecteurs propres tournent).

A un instant  $t$ , on considère donc l’ensemble des lignes que l’on a dessinées à l’étape  $t - 1$ . Pour les deux extrémités de chaque ligne, on regarde si la déformation est devenue trop faible ou si l’orientation des vecteurs propres a changé. Dans ce cas, on la raccourcit. Lorsque l’on a fini cette première passe, on retraite l’ensemble des lignes pour voir si on doit les rallonger. Finalement, on regarde si de nouvelles lignes doivent être créées. L’ensemble du processus se

résume par le pseudo-code suivant :

Algorithme d'animation des lignes de plis	
$L \leftarrow L_{prec}$	# On met dans $L$ l'ensemble des lignes de l'image précédente
Pour chaque ligne $l$ dans $L$	
	Pour chaque point $p$ extrémité de $l$
	Si $\text{déformation}(p) < \text{seuil}$ ou orientation différente de la déformation
	Raccourcir $l$
Pour chaque $l$ dans $L$	
	Pour chaque point $p$ extrémité de $l$
	Tant que $\text{déformation}(\text{extrémité}) > \text{seuil}$
	Allonger $l$ et modifier extrémité
Tant que il reste des zones déformées sans lignes	
	Créer ligne à partir d'un point initial

Un commentaire s'impose par rapport à cette méthode : pourquoi ne pas réunir les étapes 1 et 2 en une seule ? Dessiner une ligne a pour effet d'absorber la déformation dans son voisinage (rappelons que le voisinage est basiquement l'ensemble des triangle traversés par la ligne). Autrement dit, on considère qu'à partir du moment où cette ligne est tracée, aucune autre ne va venir se former dans son voisinage proche. En réunissant les deux premières étapes, une ligne pourrait s'étendre là où se trouvait un autre pli à l'image précédente. Ce pli disparaît alors subitement, engendrant un effet de popping. On pourrait certes laisser coexister les deux, mais nous n'aurions alors plus aucun contrôle sur la densité, c'est-à-dire que nous aurions plusieurs plis qui se superposent. La stratégie présentée ici a l'avantage de maintenir une cohérence d'une image à l'autre tout en évitant que des plis viennent s'agglomérer à un endroit.

Notons qu'une ligne peut être raccourcie puis allongée dans le même pas de temps. Ceci se produit dans le cas où l'orientation locale du champ de déformation se modifie alors que l'intensité reste suffisante pour y dessiner des lignes. Dans ce cas, on raccourcit d'abord notre ligne par l'étape un, puis on la rallonge mais dans une direction différente dans l'étape deux.

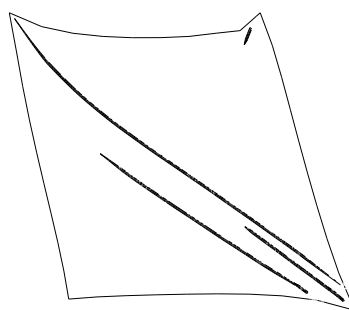
## 3.5 Rendu

### 3.5.1 Rendu calligraphique simple

Pour obtenir un rendu de ces lignes, il nous faut effectuer une étape supplémentaire. Dans le cas l'approche par triangles, nous disposons d'une suite de segments dont on voit clairement les discontinuités. Nous effectuons donc un lissage par spline cubique de ces brisés. Ces lignes lissées, ainsi que la silhouette, sont ensuite affichées en utilisant un rendu calligraphique, qui consiste à imiter le coup de pinceau d'un artiste. Concrètement, on part d'une ligne à laquelle on donne une épaisseur à l'aide de bandes de triangles. On fait ensuite varier l'épaisseur de manière à donner un effet de rétrécissement aux extrémités du trait. La figure 3.19 montre deux exemples de ce rendu.



(a) Cylindre



(b) Drap

FIG. 3.19 – Exemples de rendu

### 3.5.2 Carte de déplacement

Utilisé isolément, le rendu calligraphique simple ne donne aucun effet de drapé. Ceci est particulièrement visible, dans le cas du NPR, au niveau de la silhouette qui n'est pas déformée. Pour obtenir un rendu un peu plus intéressant, on peut passer par une carte de déplacement.

Disposant d'une paramétrisation planaire de la surface du vêtement, on dessine les lignes de plis dans l'espace texture. On lisse ensuite cette image au moyen d'un noyau gaussien dont on module la moyenne et la variance en fonction de la déformation locale de la surface. La texture ainsi obtenue permet finalement soit de donner une illusion de plis par *bump mapping*, soit de venir déformer la géométrie par *displacement mapping*.

Dans la cadre du NPR, on préférera la deuxième solution. En effet, l'artefact classique du *bump mapping* est qu'il ne modifie pas la silhouette. Celle-ci étant mise en évidence dans un rendu au trait, le défaut se retrouve accentué. On peut cependant envisager une approche mixte. A une phase de *bump mapping*, on ajoute une étape de déformation géométrique du maillage uniquement au niveau des intersections plis-silhouette où on effectue une étape de déplacement.

Précisons que ceci reste théorique car non-implémenté.

# Chapitre 4

## Bilan

A la suite de quelques exemples illustrant notre algorithme, ce chapitre discute de l'intérêt de notre méthode et de ses limitations. J'évoquerai enfin quelques pistes pour la continuation de ce travail.

### 4.1 Résultats

Les suites d'images qui servent ici d'exemples permettent d'avoir un aperçu des résultats de notre système.

La figure 4.1 montre l'animation d'un drap agité par le vent.

La figure 4.2 montre l'animation d'un cylindre qui se plie.

La figure 4.3 montre l'animation d'un tissu qui tombe sur une sphère.

Ces trois exemples illustrent trois types de déformation qui peuvent survenir dans un drapé : l'oscillation d'un tissu plat (cape ou rideau), la déformation d'une forme cylindrique (majorité des vêtements : manches, T-shirt ...) et la collision entre un tissu et un corps de forme caractéristique.

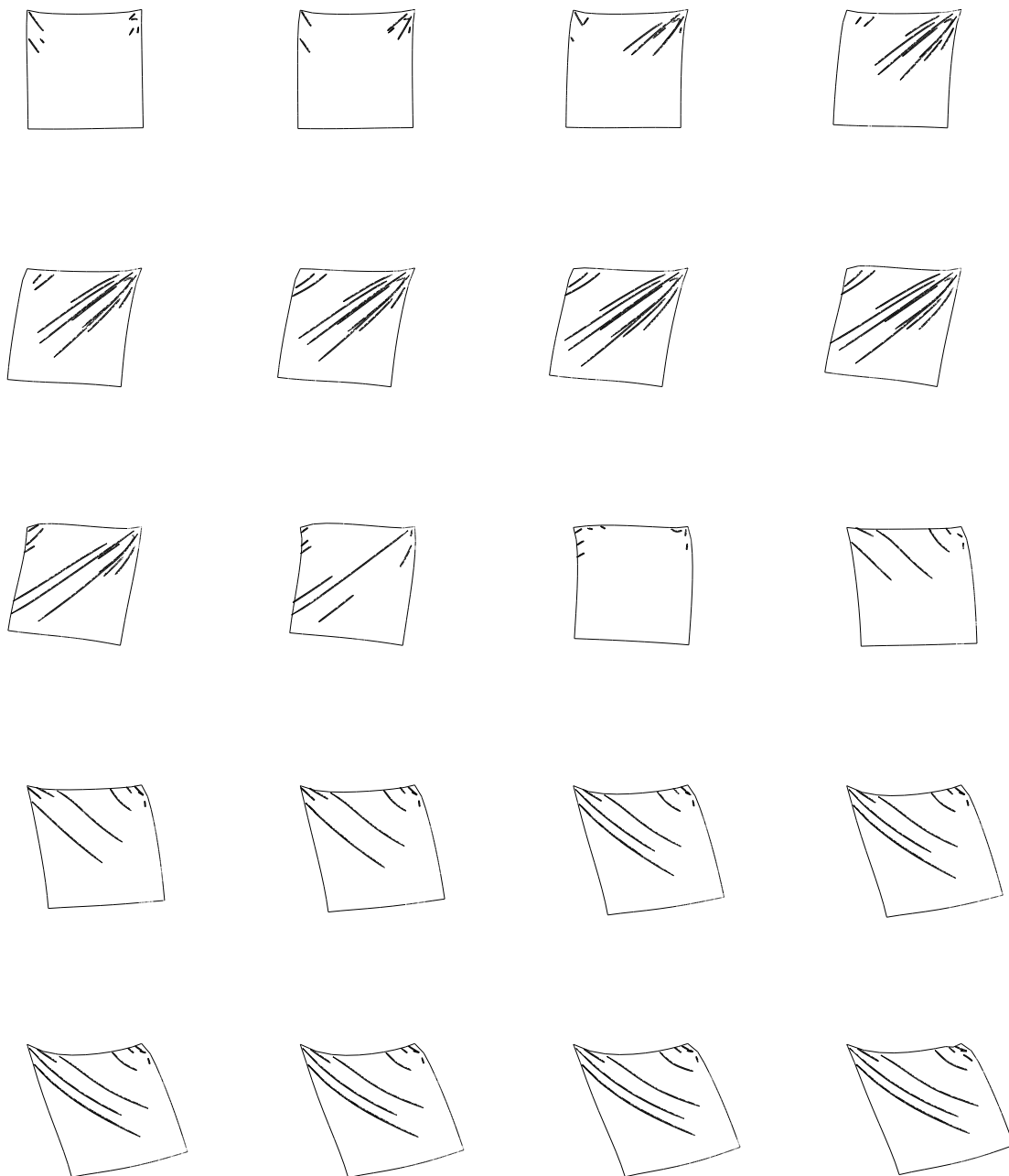


FIG. 4.1 – Exemple d'animation d'un drap



FIG. 4.2 – Exemple d'animation d'un cylindre

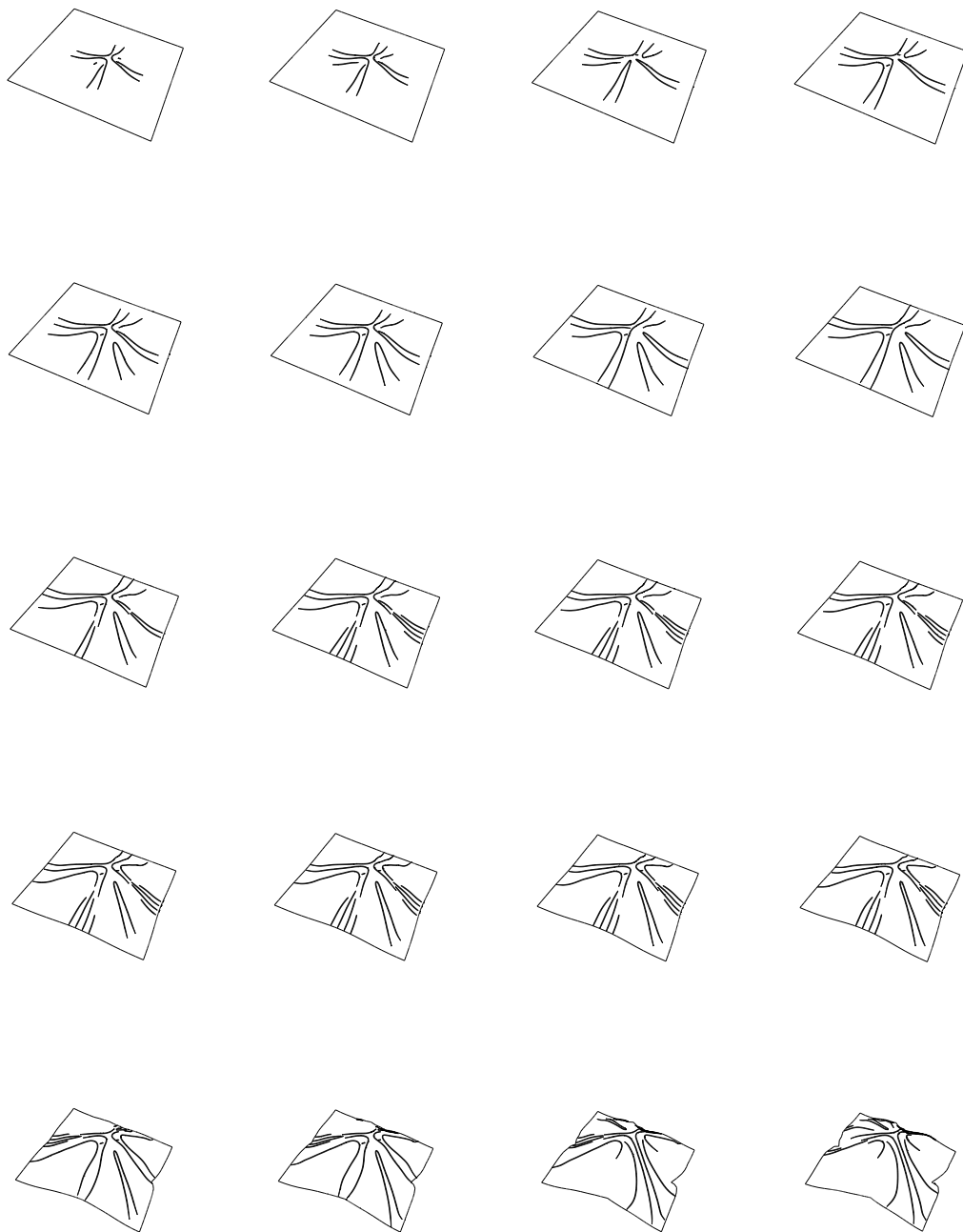


FIG. 4.3 – Exemple d'un tissus tombant sur une sphère



## 4.2 Bilan

La première constatation est que les lignes que nous traçons semblent être de bonnes lignes. Par bonnes lignes, je veux dire qu'elles paraissent donner une bonne impression de drapé, même si elles ne correspondent pas exactement à ce que l'on obtiendrait avec une simulation physique rigoureuse. Cependant, je fais partie de ceux qui pensent que la synthèse d'images s'apparente plus à l'illusion qu'à la reproduction. En ce sens, l'important n'est alors pas de fournir un résultat exact mais un résultat visuellement correct. Si le spectateur a l'impression que les plis dessinés sont bons, le but est atteint.

Ce travail montre que notre indicateur, à savoir les vecteurs propres de déformation, semble pertinent pour déterminer l'orientation des plis. Cependant, il reste probablement des recherches à effectuer sur la meilleure manière d'utiliser ce champ. Dans ce rapport, nous utilisons de manières séparées compression et extension. Peut être serait il judicieux de mettre en place une méthode qui prend mieux en compte l'interaction entre les deux directions principales de déformation.

La méthode d'animation mise en place assure une cohérence temporelle d'une image à l'autre. Nos plis bougent, s'allongent et se raccourcissent sans que cela soit choquant visuellement. L'algorithme est de plus rapide et générique. Rapide car il permet de calculer plusieurs images par seconde ; générique car il ne fait aucune hypothèse sur la nature du vêtement, contrairement à certaines méthodes qui marchent uniquement dans le cas de vêtements moulants par exemple. On peut penser qu'un sérieux travail d'optimisation de code permettrait de le faire fonctionner en temps réel.

Finalelement, bien que les exemples présentés dans ce rapport soit relativement simples, le champ d'application de notre méthode est quant à lui très vaste. Même si l'objectif principal était l'application au NPR, nous pouvons tout à fait utiliser notre méthode dans le cadre d'un rendu réaliste. Nous pouvons aussi envisager de l'appliquer à d'autres types de rides telles que celles de la peau (poignet ou front par exemple).

## 4.3 Perspectives

Au delà de l'amélioration du rendu qui permettrait de confirmer l'intérêt de nos résultats, plusieurs perspectives sont envisageables.

La première concerne le contrôle que l'on souhaite allouer au graphiste. Si notre méthode permet aujourd'hui de régler un certain nombre de paramètres tels que le nombre de lignes et les seuils d'apparition/disparition, elle n'est pas vraiment satisfaisante pour un utilisateur. Un graphiste préférera toujours pouvoir prédéfinir lui-même certains motifs plutôt que de jouer avec des "sliders". Mais les lignes qu'il dessine peuvent alors être complètement différentes de celles obtenues avec notre simulation.

Dans [CGW<sup>+</sup>05], Cutler *et al.* présentent un système de simulation de plis basé sur un certain nombre de positions clés dessinées par un graphiste. Le problème de ce système est son côté statique et l'obligation de définir des motifs pour obtenir des plis. Nous pensons qu'il est possible de combiner cette méthode avec l'approche automatique que nous proposons. En clair, l'utilisateur laisserait d'abord fonctionner notre algorithme qu'il corrigerait ensuite sur certaines images en choisissant de garder les lignes qu'il préfère tout en ajoutant celles qu'il souhaite. Une telle méthode, combinant approches manuelle et automatique, présenterait un gain de temps appréciable pour le graphiste tout en lui laissant une certaine liberté.

On peut aussi chercher à améliorer le dessin automatique de nos lignes. Une piste intéressante est proposée par Kunii dans [KG90]. Il définit les motifs de plis comme un graphe dont les noeuds représentent les extrémités des plis, les arêtes représentant soit des crêtes soit des vallées. Sur cette idée, on pourrait modifier notre méthode de manière à calculer non plus un ensemble de lignes mais un graphe représentant les plis. Ceci soulève cependant d'autres problèmes, à savoir comment faire vivre les noeuds du graphe (la méthode de Kunii était statique à ce niveau), et comment décider si l'arête est de type crête ou vallée.

Dans [CMT02], Philippe Cordier introduit l'idée de segmenter le maillage du vêtement en différentes parties. Ces différents morceaux sont classifiés en trois catégories (serré, flottant, ou intermédiaire), à chaque catégorie correspondant une technique d'animation particulière. Bien que cette méthode soit initialement conçue pour animer le vêtement de manière globale, on peut envisager de l'adapter à la simulation de plis. Dans un tel système, chaque partie serait alors dotée d'une procédure décrivant comment les plis se forment en fonction de la déformation.

# Bibliographie

- [AT99] Christophe Arleston and Didier Tarquin. Les pétaures se cachent pour mourir. *Lanfeust de Troy*, 1999.
- [BFA02] Robert Bridson, Ronald Fedkiw, and John Anderson. Robust treatment of collisions, contact and friction for cloth animation. In *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 594–603, New York, NY, USA, 2002. ACM Press.
- [Bli78] James F. Blinn. Simulation of wrinkled surfaces. In *SIGGRAPH '78 : Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, pages 286–292, New York, NY, USA, 1978. ACM Press.
- [BMF03] R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In *SCA '03 : Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 28–36, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [BW98] David Baraff and Andrew Witkin. Large steps in cloth simulation. In *SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54, New York, NY, USA, 1998. ACM Press.
- [BWK03] David Baraff, Andrew Witkin, and Michael Kass. Untangling cloth. *ACM Trans. Graph.*, 22(3) :862–870, 2003.
- [CGW<sup>+</sup>05] Lawrence D. Cutler, Reid Gershbein, Xiaohuan Corina Wang, Cassidy Curtis, Erwan Maigret, Luca Prasso, and Peter Farson. An art-directed wrinkle system for cg character clothing. In *SCA '05 : Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 117–125, New York, NY, USA, 2005. ACM Press.
- [CK02] Kwang-Jin Choi and Hyeong-Seok Ko. Stable but responsive cloth. In *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 604–611, New York, NY, USA, 2002. ACM Press.
- [CMT02] Frederic Cordier and Nadia Magnenat-Thalmann. Real-time animation of dressed virtual humans. *Comput. Graph. Forum*, 21(3), 2002.
- [CMT04] Frederic Cordier and Nadia Magnenat-Thalmann. A data-driven approach for real-time clothes simulation. *pg*, 00 :257–266, 2004.
- [Coo84] Robert L. Cook. Shade trees. In *SIGGRAPH '84 : Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 223–231, New York, NY, USA, 1984. ACM Press.

- [CP05] Frédéric Cazals and Marc Pouget. Topology driven algorithms for ridge extraction on meshes. Rapport de recherche 5526, INRIA, 2005.
- [DFR04] Doug DeCarlo, Adam Finkelstein, and Szymon Rusinkiewicz. Interactive rendering of suggestive contours with temporal coherence. In *Third International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 15–24, June 2004.
- [DFRS03] Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. Suggestive contours for conveying shape. *ACM Transactions on Graphics*, 22(3) :848–855, July 2003.
- [DJW<sup>+</sup>06] Philippe Decaudin, Dan Julius, Jamie Wither, Laurence Boissieux, Alla Sheffer, and Marie-Paule Cani. Virtual garments : A fully geometric approach for clothing design. *Computer Graphics Forum (Eurographics’06 proc.)*, 25(3), sep 2006.
- [DKT98] Tony DeRose, Michael Kass, and Tien Truong. Subdivision surfaces in character animation. In *SIGGRAPH ’98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 85–94, New York, NY, USA, 1998. ACM Press.
- [GG01] Bruce Gooch and Amy Gooch. *Non-Photorealistic Rendering*. A. K. Peters, Ltd., Natick, MA, USA, 2001.
- [HB00] Donald H. House and David E. Breen, editors. *Cloth modeling and animation*. A. K. Peters, Ltd., Natick, MA, USA, 2000.
- [HBVMT99] Sunil Hadap, Endre Bangerter, Pascal Volino, and Nadia Magnenat-Thalmann. Animating wrinkles on clothes. In *VIS ’99 : Proceedings of the conference on Visualization ’99*, pages 175–182, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.
- [KDMF03] Robert D. Kalnins, Philip L. Davidson, Lee Markosian, and Adam Finkelstein. Coherent stylized silhouettes. *ACM Trans. Graph.*, 22(3) :856–861, 2003.
- [KG90] T. L. Kunii and H. Gotoba. Modeling and animation of garment wrinkle formation processes. *Proc. Computer Animation ’90*, pages 131–147, 1990.
- [NPF05] Matthieu Nesme, Yohan Payan, and François Faure. Efficient, physically plausible finite elements. In J. Dingliana and F. Ganovelli, editors, *Eurographics ’05 (short papers)*, august 2005.
- [OBS04] Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph.*, 23(3) :609–612, 2004.
- [OM01] Masaki Oshita and Akifumi Makinouchi. Real-time cloth simulation with sparse particles and curved faces. In *Proceedings of Computer Animation*, November 2001.
- [SS02] Thomas Strothotte and Stefan Schlechtweb. *Non-photorealistic computer graphics : modeling, rendering, and animation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.
- [VMT99] Pascal Volino and Nadia Magnenat-Thalmann. Fast geometrical wrinkles on animated surfaces. In *Seventh International Conference in Central Europe on*

*Computer Graphics and Visualization (Winter School on Computer Graphics)*, February 1999.

- [VMT01] Pascal Volino and Nadia Magnenat-Thalmann. *Virtual Clothing with Cdrom*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
- [YBS05] Shin Yoshizawa, Alexander Belyaev, and Hans-Peter Seidel. Fast and robust detection of crest lines on meshes. In *ACM Symposium on Solid and Physical Modeling*, pages –, MIT, USA, 2005. Association for Computing Machinery (ACM), ACM.